06-01-00

A

THE ASSISTANT COMMISSIONER OF PATENTS
Washington, D.C. 20231

DOCKET NUMBER: AUS000123US1
May 31, 2000

Sir:

Transmitted herewith for filing is the Patent Application of:

Inventor: BRUCE A. BEADLE, ET AL

For: DYNAMICALLY AFFECTING BROWSER NETWORK COMMUNICATIONS PERFORMANCE

Enclosed are:

X Patent Specification and Declaration

X 8 sheets of drawing(s).

X An assignment of the invention to International Business Machines Corporation (includes Recordation Form Cover Sheet).

__ A certified copy of a __ application.

__ Information Disclosure Statement, PTO 1449 and copies of references.

The filing fee has been calculated as shown below:

For	Number Filed	Number Extra	Rate	Fee
Basic Fee				$690
Total Claims	29	-20	9	x 18 = $162
Indep. Claims	4	- 3	1	x 78 = $78
MULTIPLE DEPENDENT CLAIM PRESENTED				x260= $
			TOTAL	$930

X Please charge IBM Corporation Deposit Account No. 09-0447 in the amount of $930. A duplicate copy of this sheet is enclosed.

X The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to IBM Corporation Deposit Account 09-0447. A duplicate copy of this sheet is enclosed.

X Any additional filing fees required under 37 CFR §1.16.

X Any patent application processing fees under 37 CFR §1.17.

CERTIFICATE OF MAILING BY "EXPRESS MAIL" UNDER 37 CFR § 1.10

"Express Mail" mailing label number EL453461416US

Date of Mailing May 31, 2000

I hereby certify that the documents indicated below are being deposited with the United States Postal Service under 37 CFR 1 10 on the date indicated above and are addressed to Box Patent Applications, Assistant Commissioner of Patents, Washington, D C 20231 and mailed on the above Date of Mailing with the above "Express Mail" mailing label number

Barbara Spence Barbara Spence

Respectfully submitted,

Eustace P. Isidore

By Eustace P. Isidore
Registration No. See Attached
FELSMAN, BRADLEY, VADEN, GUNTER & DILLON, LLP
Suite 350 Lakewood on the Park
7600B North Capital of Texas Highway
Austin, Texas 78731
Telephone (512) 343-6116

DYNAMICALLY AFFECTING BROWSER NETWORK COMMUNICATIONS PERFORMANCE

RELATED APPLICATIONS

The present invention is related to the subject matter of the following commonly assigned, co-pending United States Patent Application Serial No. _____ (Docket No. AUS000229US1) entitled "Dynamic, Seamless Switching of a Network Session from one Connection Route to Another" filed concurrently herewith.

BACKGROUND OF THE INVENTION

1. Technical Field:

The present invention relates in general to computer networks and in particular to communications within computer networks. Still more particularly, the present invention relates to a method, system, and program product for improving a client's communication performance within a network by automatically selecting a best performing network route for the client communication.

2. Description of the Related Art:

Computer networks, such as the Internet, are typically based on Client/Server software architectures. These architectures provide a versatile infrastructure that

supports shared access to server resources. A client is typically a requester of services, and a server is the provider of services. A single machine can be both a client and a server depending on the software configuration. A client machine is loaded with client software, while a server machine is loaded with server software. Clients can be either stand-alone computer systems (like personal computers) or "dumber" systems adapted for limited use with a network.

A generalized client-server computing network has several nodes or servers which are interconnected, either directly to each other or indirectly through one of the other servers. Based on network configuration, the connection by the client to the network may be via an ethernet connection or token ring, etc. Other direct and/or indirect connection methods (e.g. telephone connection via remote access protocol) are also possible when a client is connected from the user's home, for example. In traditional networks, only a single connecting medium to the network is possible for each client.

The network can be a localized network or geographically dispersed network and can be further connected to other networks. Each server is essentially a stand-alone data processing (computer) system, having one or more processors, memory devices, and communications devices, that has been adapted (programmed) for providing information

and/or services to individual users at another set of nodes or client workstations.

In a Local Area Network (LAN), for example, clients are usually configured to communicate via a particular server. For geographically dispersed networks, as in an intranet or Internet, a client's network communication may occur via a localized server or a server in another geographic location. The use of auto-proxy servers in a LAN system helps to improve network performance. However, only one connection route may be configured within the client at a particular time and these servers can become congested during heavy usage or traffic. Also, current client browsers (i.e., software applications for accessing and communication with the Internet) are typically configured to transmit network traffic via a specific server and once configured, remain set in that configuration until the user manually changes the settings to utilize another server. Thus, if the server connection is congested, the user may either remain on the connection or terminate the session and attempt to secure a faster connection by re-connecting to the same server. Also, if the connection is lost during a session, the session information is usually lost (or unrecoverable) and the user has to re-connect via the same server and start the session over.

Network input/output (I/O) performance is often a problem when using a browser in an intranet to access the World Wide Web or Internet. In most intranet cases, the

client is required to use a sock server or a proxy server before a connection can be made to the Internet. In a typical intranet, there are several sock or proxy severs that may be geographically dispersed within the network. The performance of the servers varies throughout the day due to factors such as location, time of day, loading, etc. Thus, in most networks, auto-proxy servers, which filter the local intranet traffic from the external Internet traffic, are utilized to improve performance of sock servers.

To address such performance concerns, special tools or components have been developed to improve communication to servers connected to the Internet. For example, various load balancing solutions have been implemented, such as Local Director of Cisco Systems. Local Director performs load balancing and interfaces with the Internet-based client attempting to access the content servers. All traffic across the Internet to content servers is thus shielded by Local Director. U.S. Patent No. 5,774,660 discloses a load balancer having a virtual address connected to a multi-node server. The load balancer receives all incoming Internet requests for server access, and modifies the destination address to reflect the correct server node that ultimately services the request. These solutions are all server-based and implemented on the back end of the network, i.e., the server connection to the Internet itself.

The present invention recognizes that it would therefore be desirable to have a method, system, and program

product for dynamically improving the network access and communication performance of a client. A method, system, and program product, which allows a client browser to automatically select a best connection route for communicating to both intranet and Internet servers would be a welcomed improvement. It would be further desirable if such a method, system, and program product permitted a single client to select between multiple connection media based on a connectivity history to select the optimal route for connecting to a network. These and other benefits are provided in the present invention.

SUMMARY OF THE INVENTION

A browser application for utilization within a client system that provides an optimal connection to a network is disclosed. The browser application comprises a graphical user interface for receiving user selection and connection requests, and a connection utility for connecting the client system with a selected server of the network. The selected server is automatically selected from a plurality of servers in response to the receipt of a connection request, based on a determination of a best route for completing the connection request.

In one embodiment of the invention, a single client is provided with multiple connection media and the determination of the optimal route includes a selection of the connection medium to utilize for the connection. Also, a table of connection information is stored within the client and utilized in evaluating which connection route is optimal. The determination of the best route may be based on historical data stored within a table or current analysis of the network. Network information is obtained by encoding call-backs within the header of the address protocol, which monitors the network routes and transmits relevant connection information to the client system.

The above as well as additional objectives, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives, and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 depicts a data processing system, in which a preferred embodiment of the present invention may be implemented;

Figure 2 is a high-level block diagram of a client browser within an intranet configuration in accordance with one embodiment of the present invention;

Figure 3 illustrates a browser-network configuration with multiple selectable server-connecting media and/or routes in accordance with a preferred embodiment of the present invention;

Figure 4 is a block diagram of a modified browser application with a selectable connection button for opening a connection settings window in accordance with a preferred embodiment of the present invention;

Figure 5A is a block diagram of a server-connections' GUI utilized in accordance with a preferred embodiment of the present invention;

Figure 5B is a block diagram of an in-session connection modification window utilized in accordance with a preferred embodiment of the present invention;

Figure 6A is a block diagram of a table of connection attributes utilized in accordance with a preferred embodiment of the present invention;

Figure 6B is a flow chart of the process of determining a best connection route in accordance with a preferred embodiment of the present invention;

Figure 7 is a flow chart of the process of completing a network connection in accordance with a preferred embodiment of the present invention;

Figure 8 is a flow chart of the process of switching a session from one connection route to another in accordance with another preferred embodiment of the present invention;

Figure 9 is a block diagram of the TCP/IP header with an encoded call-back component for directing client call-backs in accordance with a preferred embodiment of the present invention; and

Figures 10 and **11** are block diagrams of specific examples of a client connected to a network via two possible routes to enable the switching of a session from one connection route to another in accordance with another preferred embodiment of the present invention.

DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

The present invention is directed to a method, system, and program product implemented within a client system, which allows a browser application to automatically select an effective network route from the client system to a network during network communication. As utilized within the invention, the term "effective" refers to a better performing (if not the best performing or optimal) route, based on known or available criteria and present network conditions. The preferred embodiment is described with reference to optimal routes, but is done so merely for illustrative purposes. In particular, a software implemented application of the present invention expands the functionality of a traditional browser to permit a multiple-route connection setting (or configuration), by which the browser is not restricted to a connection via a single specific server and connection medium, but is permitted to select from among multiple servers and multiple connection media. The software implemented application is referred to herein as a connection utility. In a preferred embodiment, the invention reduces the need to manually change a network communication configuration from a particular server to another and enables the browser to have flexibility to ensure the best or most efficient connection routes possible under current loading conditions.

The invention is implemented in the data processing system and network environment as illustrated in **Figures 1**

and **2** and **3**, respectively. The invention may, however, be implemented in other types of data processing systems and networks, so, while the present invention may be described with references to these figures, these references should not be construed in a limiting sense.

For the purposes of this invention, the term client is utilized to refer to both the hardware component which is connected to a network server and the software applications stored in memory and being run on the hardware component. Also, the term client is utilized to mean both a directly connected network computer system and a stand-alone computer system, such as a user's home-based computer system. In the preferred embodiment, a client is provided with browser application and connection utility for accessing the Internet via several possible connection routes. User refers primarily to an individual who utilizes the client. Further, the invention applies to connections to servers of both an intranet and Internet via the browser application running on the client.

The servers of the network are typically data processing systems having a database, operating system (OS), and server software. The server software operates within a network server and provides the support for Internet access by clients.

With reference now to the figures and in particular with reference to **Figure 1**, a computer system that may be

utilized as a standalone computer system or one of the client or servers on a network is presented. Computer system **20** comprises a Central Processing Unit (CPU) housed in a system unit **22**. System unit **22** also provides connections for various hardware components including disk drives **40** and memory devices (not shown). Several Peripheral input/output devices are connected to the CPU. These input/output devices include keyboard **82**, mouse **84**, printer **94**, CD ROM **78**, and display monitor **30**.

Also coupled to CPU are various external devices, including modem **92**, and network adapter **90**, utilized for connecting data processing system **20** to other systems and/or networks, as is illustrated in **Figures 2** and **3**. CD rom **78**, modem **92**, and network adapter **90** are depicted as external components; however those skilled in the arts are familiar with the various architectures of data processing systems and understand that these components may be housed inside of system unit **22**.

Modem **92** is a communication device that enables computer system **20** to transmit information over standard telephone lines or wireless connections such as cellular. Modem **92** converts digital computer signals to interlock signals suitable for communications over these telephone media. Modem **92** can be utilized to connect data processing system **20** to a server via remote access protocols. Modem **92** may also provide a connection to other sources, such as an

electronic bulletin board (BBS) or the World Wide Web (Internet). Network adapter **90** may be used to connect data processing system **20** to a network such as an intranet as depicted in **Figure 2**.

Computer system **20** also preferably includes an interface such as a graphical user interface (GUI) provided by an operating system (OS) **24** that resides within machine readable media to direct the operation of computer system **20**. Any suitable machine-readable media may retain the OS, such as RAM, ROM, SCSI disk drive, and other disk and/or tape drive(e.g. magnetic diskette, magnetic tape, CD-ROM, optical disk, or other suitable storage media). Also, any suitable OS **24** may direct the CPU of the data processing system **20**.

Further, computer system **20** preferably includes at least one software application (program product) **26** that resides within machine readable media. The software application may be accessible by OS **24**, or may be incorporated into an OS control program. Preferably, software application **26** contains instructions that when executed on the CPU carry out the particular operations of the present invention as described herein.

SELECTING BROWSER-NETWORK CONNECTION

Referring now to **Figure 2**, a high-level block diagram of an intranet with multiple server connections for a single client is illustrated in accordance with a preferred embodiment of the present invention. Intranet **200** comprises a plurality of servers that are at different geographical locations. These servers include Chicago servers **203A**, New York servers **203B**, California servers **203C**, and Texas servers **203D**. Client browser **201** connects to each server via a different connection medium. Thus, client browser **201** is connected via ethernet **205** to Chicago servers **203A**, via Token Ring **209** to New York servers **203B**, via an internet Service Provider (ISP) to California servers **203C**, and via Satellite **211** to Texas servers **203D**. In a preferred embodiment, the ability of a client to connect to different servers using different connecting medium is a major consideration utilized in selecting optimal routes as will be described below. The invention, however also finds applicability in a network in which each server is connected to the client via the same connecting medium. Intranet **200** connects to Internet (not shown) via network Internet connectors **215**. Client browser access to the Internet is routed through the Intranet servers **203A-203D**, which in effect serve as proxy servers.

The preferred embodiment of the present invention places the capability to choose the best performing network route in the client (i.e., away from the server) and specifically within the browser or other network access

application running on the client. The invention assists in solving the congestion problems within an Intranet and/or the Internet. The invention also involves the encoding of software call-backs into the protocol stack and executing of a software application that utilizes the call-back utility to assist in choosing a better performing network route. The invention has greatest applicability in an intranet and/or Internet environment, but may also apply to basic LANs or other types of multiple-server networks.

For the purposes of the invention, use of the term network is understood to refer to both Wide Area Netwroks such as the Internet and to Local Area Networks (LAN) such as an intranet. The network types may provide different access methods but are essentially similar in their functionality, as those skilled in the art are aware. A network preferably consist of a number of servers and provides network clients with a common protocol for communicating and transferring software and information electronically within the network environment.

Turning now to **Figure 3**, there is illustrated another network configuration including a client browser coupled to multiple servers. The illustrated network may represent an intranet configuration as in **Figure 2** or a stand alone client configuration. The intranet configuration shows a client, connects to a server of the intranet. The intranet server is then utilized to provide access to Internet servers. In the stand alone client configuration, a client

connects directly to a server of the Internet (i.e., without the intranet server). As an Intranet configuration, the elements of **Figure 3** are similar to the elements of **Figure 2**, but client may be connected utilizing remote access (e.g., access from a user's home). However, in a stand-alone client configuration, browser **301** is not coupled to an Intranet. Servers **303A-303D** may be servers of different Internet Service Providers (ISPs), which are connected to browser **301** via different connection media. Thus, browser **301** is connected to server **303A** via a DSL connection **305**, to server **303B** via a wireless phone connection **307**, to server **303C** via a satellite connection **311** and to server **303D** via a cable modem connection **309**. ISP servers **303A-303D** connect to the Internet (not shown) via ISP-Internet connectors **215**.

The implementation of multiple connection media for a single client is further enhanced by permitting a user to optionally select to utilize a particular routing medium based on the desired speed of connection. Thus a satellite connection may be selected for huge data transfer, while a DSL or phone modem connection is utilized for email correspondence. The selection of a particular medium becomes even more important when each medium has a corresponding cost factor, which directly affects the user's determination of use. In another embodiment, a single server may be accessible to the client browser using one of multiple connecting media. Thus for example, servers **303A-**

303D may represent a single server having four ports that are each associated with a particular medium.

Figure 4 illustrates a GUI of a client browser application utilized in a preferred embodiment of the invention. Client browser **400** is the browser application running on a client, which controls connection to the network. In the preferred embodiment, client browser **400** has selection buttons **401,** a search field **403,** display frame **405**, etc. Client browser **400** also has a "connection" button **407**, which when selected by a user opens the connection settings GUI illustrated in **Figure 5A**.

Referring now to **Figure 5A,** there is illustrated a connection settings GUI according to a preferred embodiment of the present invention. Connection settings GUI **500** has a default selections frame **501** and a server selections frame **503**. Default selections frame **501** contains a select button for each connection medium available to the client system. In the illustrative embodiment, select buttons for DSL (i.e., phone-based connection) **505A**, wireless **505B**, Satellite **505C**, and cable modem **505D**, are illustrated. The presently available connection media are represented with full lines, while the presently unavailable connection medium (i.e., satellite) is represented with dashed lines. Also, the currently selected medium is highlighted.

Server selections frame **503** allows the user to select an option for determining which connection routes to utilize during server connections. Three major routing options, presented as selectable buttons, are available. The selectable buttons of the routing options include select default server button **507**, override defaults button **511** and automatic routing button **513**. Selection of select default server button **507** (or override defaults button **511**) causes a pop-up menu **509** of possible server connections to be displayed. A user may thus select a particular server as the default server and force all connections to be routed to that particular server. When the override defaults button **511** is selected, however, the pop-up menu **509** is utilized to select a primary server and alternate option(s) for routing the connection when the connection via the default server is bad or congested, etc. In a preferred embodiment, when an alternate server route is determined to be more optimal than the default server route, the current default route is automatically replaced within the connections settings by the alternate route. The alternate route is stored by the connection utility as the new default route for use during later connections.

Once the user has completed his selections, he may select the Update button **515**, which updates the default connections and server selections prior to exiting connection settings GUI **500**. The user may also select the

exit button **517** which exits the GUI **517** without changing the previous settings.

• Selection of automatic routing **513** enables the browser to access a stored connectivity table, illustrated in **Figure 6A**, which stores specific routing information by which the optimal route for a particular connection is determined. In the preferred embodiment, selection of a best performing network route involves utilizing the connectivity table **600**. The connectivity table **600** is utilized to track and store a history of network and routing parameters **601**, which indicate the efficiency of a particular server or route using efficiency values ranging from, for example, 1-5. The network parameters are generated by first encoding software call-backs into the connection protocol stack and creating within the connection utility a routing function, by which these software call-backs may transmit network routing information back to the client. The functionality and utilization of software call-backs in network applications is described in detail in United States Patent Application Serial No. _____ (Docket No. AUS000122US1) entitled "Extending Functionality of Network I/O APIs by Using Call-Back Mechanisms," which is filed concurrently herewith and hereby incorporated by reference. Initial parameter values may be set by default or by a user; However, all subsequent values are set utilizing the call-back functionality. The parameter values may be updated

during each connection request or at some pre-determined period, for example, every 6 hours.

Figure 9 illustrates a Transmission Control Protocol/Internet Protocol (TCP/IP) header for an Internet connection protocol. TCP/IP header **900** contains protocol stack **901** that is utilized to route client requests via web browser. Protocol stack **901** of TCP/IP header **900** is modified to include a set of call-back registers **903,** which complete the monitoring of current network connection and extracts relevant parameter data (or metrics). The set of call-back registers are programed to return the parameter data to the client prior to termination of the network connection.

Returning now to **Figure 6A**, connectivity table **600** contains a mapping of servers (or connection routes) **603** presented in columns with a list of corresponding filters (or parameters) values **601,** which are utilized to determine a relative connection rating of each of servers **603**. The illustrated parameters values (metrics) **601** includes location of server (i.e., with respect to the web site, since a server tends to have quicker connection speeds to a web site that is closer than one at a further distance), connecting medium bandwidth, latency, packet size, reliability, server capacity, and cost. Other parameters may be included, for example, relative security and time of day (e.g., for clients in the eastern U.S., in the early

morning, servers in the western U.S. may be less congested than servers in the Eastern U.S. because of the time difference).

Figure 6A provides a specific 2 domain embodiment. A domain is an IP address usually associated with a server on the Internet. In **Figure 6**, for example, ibmgame.com domain represents a server, which hosts an interractive game that is accessed by the client browser (i.e., the user of the client browser). The server for ibmgame.com domain is accessed from the client via a particular route and the associated metrics of that route are stored within the connectivity table **600**. The columns illustrate three pairs of connection parameter values. A relative rating **605** of the pairs utilized in the determination of the optimal route.

The first pair represents the user preferences for each parameters when connecting to both general domains (i.e., other domains besides the ibmgame.com domain) and when connecting to ibmgame.com domain. These settings are made with the default selection of connection settings GUI of **Figure 5A**. Each parameter is assigned a value ranging from 1 to 5, with 1 representing a good (or high priority or important) metric and 5 representing a bad (or low priority or unimportant) metric. For, example, in the first column pair, latency has a relatively high priority when connecting

to the ibmgame.com domain. Bandwidth, however, has a low priority.

The second column pair indicates the average metrics for each parameter when connecting to server A and server B. Accordingly, server A provides the best connection to the general domains. The third column pair illustrates the metrics for connecting to ibmgame.com domain for each server. As illustrated, server B provides relatively better connections to this domain than does server A. Thus when a user wishes to connect to a general domain, the connection utility automatically selects server A to route the connection, unless the user specifically request connection via server B. However, when the user desires to connect to ibmgame.com domain, the connection utility automatically routes the connection via server B, based on the aforementioned metrics.

Figure 6B illustrates one implementation of a process in which the routing information from the connectivity table of **Figure 6A** is utilized. The process begins at block **651** and thereafter proceeds to block **653**, where the user preferences are obtained. A determination of whether historical data on the metrics of the connection exists is then made at block **655.** If there is historical data, a determination is made whether the user preferences indicate that the user desires to utilize the historical data depicted at block **657.** If the user wishes to utilize the

historical data, the best guess results are then computed utilizing the historical data from the database and the user preferences at block **657**. The result is then used as the basis for the connection. If, however, there is no historical data or the user does not wish to utilize the historical data, new metrics are generated from the current connections at block **659**. Thus, the user may decide if he wants to accept an immediate best guess, which is almost immediate and based on historical performances (i.e., the user preference plus the previously recorded metrics) or alternatively if he wishes the connection utility to complete an actual evaluation of current conditions. The later method will result in some delay, which may be noticeable based on the speed of the hardware being utilized.

Once the new metrics are generated, they are compared against the user's preferences to determine the optimal connection at block **661**, and the session is routed to the optimal connection at block **663**. The connection utility then monitors the connection for changes in the attributes of the server, connection, or the user preferences at block **665**. A change in a metric may result in the process returning to block **651** and similar analyses being conducted until the session is terminated. The process then ends at block **667**.

Since each client connects to only a limited number of servers, the actual table is held to a manageable size and updated after each connection or series of connections without utilizing a lot of the client's memory capacity. The function which computes the relative ratings of each connection type is linked to the connection utility to provide a resulting best connection route. The connection utility then automatically updates the TCP/IP header with the routing information. Thus, when a connection is requested, the connection is automatically routed to the optimal route without requiring user input.

Referring now to **Figure 7**, there is illustrated another embodiment of the network connection process beginning at block **701** and thereafter proceeding to block **703**, which depicts the user of the client requesting an Internet connection. The connection request triggers the connection utility, which determines if default server connections were set at block **705**. If default server connections were set, then the connection request is routed via the default server at block **707**, and the process ends at block **709**. If, however, a default server connection was not set, the connection utility next determines if an override default (i.e., default with suggestions) option was set at block **711**. If an override default option was set, the current default connection is completed at block **713**. Simultaneously, alternate routes are checked at block **715** to enable the connection utility to determine if a more optimal

route is available at block **717**. If there is a more optimal route, the browser's connection settings are updated to reflect the optimal route as the default route at block **719**. Then the process ends at block **709**.

Returning to block **711**, if default with suggestions option was not set, then the connection utility checks the connectivity table for the optimal route at block **721**. The connection is completed via the optimal route at block **723**. An update of connection status is made within the connectivity table at block **725**, and then the process ends at block **709**.

SEAMLESS SWITCHING OF A SESSION CONNECTION

In another preferred embodiment of the invention, a single client browser is permitted to simultaneously access the Internet via more than one network connection routes during a single network session. A session may consist of the transfer of multiple data types, each of which has a corresponding optimal route for transmission. The word "data" as utilized herein is defined to mean any type of information which may be transmitted over a network connection during a session. The optimal routes may be determined by factors such as the connection medium, associated cost, and level of security, as described with reference to **Figure 6A**. For example, a financial session which includes confidential or time sensitive financial information and general customer data may have the financial

information routed via a first route and the customer data routed via another route. This embodiment assumes seamless switching capability during a session and utilizes clients that are capable of simultaneously connecting to a network via multiple connection routes.

According to this embodiment and as illustrated in **Figure 10**, client browser **1001** may be simultaneously connected to a network (e.g., the Internet) via a first route **1003** and a second route **1005**. First route **1003** connects client browser **1001** to Internet **1011** via a satellite **1007** (for example, a DirectPC connection), which has high bandwidth for transfer of large files but exhibits high latency. Second route **1005** connects client browser **1001** to Internet **1011** via a phone and modem connection **1009**, which typically has low bandwidth for transfer of large files but exhibits low latency. During the session, transfer of data, such as customer data, which is not time sensitive (i.e., does not require low latency) but requires large bandwidth is completed via first route **1003**. The transfer of time sensitive financial data, which requires less bandwidth is completed via the second route **1005**. During operation, connection utility of client browser **1001** monitors the transmissions within a session to detect the data type, and controls the routing of the data via its optimal route. In one embodiment, the optimal route for the various definable data types are determined and stored in the connectivity table or other location which is accessible

by the connection utility. During connection, the connection utility monitors the network session for the occurrence of specific pre-defined characteristics, which are exhibited by particular pre-defined data types and for which network performance may be enhanced by routing the data types via a different route. Alternatively, the user may be permitted to set default routes based on data types prior to commencement of the session. Thus, although all the data is transferred to the same destination during a single session, each data type may be routed along different connection paths.

In another embodiment, in which only a single route may be connected at a given time during a session, a first portion of the session is routed via a first connection route. Whenever a measurable change occurs in the connection route, based on predefined criteria, the client dynamically routes that new data type via a next connection route. For example, if the connection is lost during a session or the connection becomes congested or too slow, a next route is automatically selected for completing the session. Selection of an alternate route may occur prior to connecting the session. The connection utility is provided with the set of criteria and the alternate routes. This information is made accessible to the connection utility. During session connection, the connection utility monitors the network session for the occurrence of the specific pre-defined characteristics, which triggers the routing of the session via a different network route.

Figure 8 illustrates the process of switching connection routes during a session in accordance with one preferred implementation of the invention. The process begins at block **801** and thereafter proceeds to block **803**, where a connection route is established. Session information is entered by a user at block **805**. During a session, the connection utility determines if a particular condition that warrants the change of the present connection route to another connection route has occurred at block **807**. The conditions which may trigger this response includes a determination of (1) a more optimal connection route (where the route is significantly more optimal relative to the present route), (2) a bad connection on a present route, (3) a loss of connection on a present route, whether due to server failure of connection medium failure, change in data type, etc. In the illustrated embodiment, the optimal route is selected at block **809** and the session switched to the optimal route at block **811**. The session is then completed at block **813** on the connected route, and the process ends at block **815**.

In the preferred embodiment, the switching step occurs without any significant loss of information from the session. Thus, in one embodiment, session information (i.e., client entered keystrokes and other events) is cached on the client system until the session is terminated by the user. If the session connection is lost or another

condition occurs, the connection utility merely reconnects via a different route and downloads the cached session information to the server. The user is thus able to continue the session from the point at which the session was interrupted. With faster connection media, such as cable modem, the switching and re-connection of a session may occur quickly enough to not interrupt the user interfacing with the server. This is referred to as seamless switching of session connection routes.

In a related embodiment, a session may be automatically re-routed to the second route only during later connection requests (i.e., the browser default is updated to select the second route). Alternatively, a "switch connection" button can be included in client browser for switching between connection states during a session. Or, in yet another embodiment, illustrated in **Figure 5B**, a pop-up window **551** is initiated whenever connection utility finds a connection route which is more optimal than the present route. The user may then elect to connect via the more optimal route by selecting a "switch" button **553** within the pop-up window.

Figure 11 illustrates a network environment in which a client session is re-routed. The network environment includes a mobile client browser **1101**, such as a browser on a laptop, hand-held personal computer or mobile telephone, which is connected to the network via a wireless connection. Client browser (or network application) **1101** has TCP/IP

stack extensions **1003**, which determine the route for connecting the client during a session. A session, comprising network packets, may initially be routed to the Internet **1007** via a wireless LAN **1109** (e.g., bluetooth), which allows wireless transmissions for significant distances. When the client moves out of range of the wireless LAN **1109**, the session is dynamically switched over to a cell phone and modem connection **1105**. Dynamic switching allows the session to be continued without any significant delay or loss in session information.

Some of the benefits and applicability of the present invention includes:

1. providing dynamic assignment of proxies, name servers, etc., according to predetermined (but programmable) performance criteria embedded in the client browser;

2. enabling a user to define multiple IP addresses for proxy or name servers rather than being restricted to a single address and related server as is currently available;

3. allowing a client browser to scan each of the entered IP addresses for the one that gives the best performance for a particular target destination;

4. allowing a client browser to skip to an alternative IP address if the current server connection is terminated or is slow to respond;

5. allowing a client browser to use more than one data path per session;

6. allowing optimum performance of the client browser, which can now interface utilizing an API architecture with a network via a back-end server of a load balancing system;

7. allowing enhanced performance by reducing the number of segments traversed between a client browser and a target server by selection of the most direct path, when that path is most optimal;

8. supporting concurrent Cable Modem, Modem, T(1-N), INTERNET2, and other connection capabilities to enable multiple open proxies and other server connections;

9. providing a user friendly interface for the above features; and

10. providing performance enhancements not limited to network connected client browsers, but configurable for any client as well as server workstations;

As a final matter, it is important that while an illustrative embodiment of the present invention has been, and will continue to be, described in the context of a fully functional data processing system, those skilled in the art will appreciate that the software aspects of an illustrative embodiment of the present invention are capable of being distributed as a program product in a variety of forms, and that an illustrative embodiment of the present invention applies equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of signal bearing media include recordable type media such as floppy disks, hard disk drives, CD ROMs, and

transmission type media such as digital and analogue communication links.

Although the invention has been described with reference to specific embodiments, this description is not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as alternative embodiments of the invention, will become apparent to persons skilled in the art upon reference to the description of the invention. It is therefore contemplated that such modifications can be made without departing from the spirit or scope of the present invention as defined in the appended claims.

CLAIMS:

What is claimed is:

1. A method for providing a client with a connection to a network, said method comprising the steps of:

selecting a connection type from a plurality of connection types; and

in response to a receipt of a connection request, dynamically connecting said client to a selected server of said network based on a determination of an effective route for completing said connection request, given said selected connection type.

2. The method of Claim 1, wherein said selecting step includes the step of providing a graphical user interface with selectable options for each of said plurality of connection types, in response to a user request to configure said client with one of said plurality of connection types.

3. The method of Claim 2, wherein said selecting step includes the step of selecting an effective server connection based on a connection history of said client and present connection conditions.

4. The method of Claim 3, wherein said selecting step includes the step of accessing said connection history from a table of server connection parameters, which are utilized to determine said effective connection route.

5. The method of Claim 4, wherein said dynamically connecting step includes the step of evaluating said server connection parameters for each of a plurality of servers to determine said effective connection route relative to all other possible routes within said connection type.

6. The method of Claim 5, wherein said dynamically connecting step further includes the step of encoding routing information about said effective connection route in a connection protocol of said client.

7. The method of Claim 6, wherein said encoding step includes the step of including a call-back mechanism in said connection protocol, wherein relevant connection information, including one or more of said connection parameters, is returned to said client for updating said table.

8. The method of Claim 7, wherein said client is equipped with multiple connection media and said dynamically

connecting step includes the step of selecting one of said multiple connection media to complete said connection request.

9. The method of Claim 8, wherein said selecting step includes the step of selecting a connecting media, which provides the effective connection route.

10. A computer program product for utilization within a client for connecting to servers of a network, said program product comprising:

a computer readable medium; and

program code on said computer readable medium, which provides:

an interface for receiving user input and connection requests; and

a connection utility for dynamically connecting said client to one of said servers in response to a connection request, wherein said one of said servers is selected based on a determination of an effective route for completing said connection request.

11. The computer program product of Claim 10, wherein program code for said interface further comprises program code for a connection selection interface for receiving user selection of a desired connection type, wherein said desired connection type is selected from a plurality of selection types including a default server connection, a changeable default server connection with a suggestion function for providing an optimal server connection during a later connection, and an effective server connection based on a connection history of said client.

12. The computer program product of Claim 11, wherein said program code for said connection utility includes:

program code for managing a connectivity table utilized to record a plurality of connection parameters for each of said servers;

program code for determining said effective route based on said connection parameters;

program code for encoding a connection protocol with said effective route; and

program code for appending a call-back to said connection protocol, whereby connection parameters from a current connection is returned to update said connectivity table.

13. A system for providing a client with a connection to a network, said system comprising:

means for selecting a connection type from a plurality of connection types; and

in response to a receipt of a connection request, means for dynamically connecting said client to a selected server of said network based on a determination of an effective route for completing said connection request, given said selected connection type.

14. The system of Claim 13, wherein said selecting means includes means for providing a graphical user interface with selectable options for each of said plurality of connection types, in response to a user request to configure said client with one of said plurality of connection types.

15. The system of Claim 14, wherein said selecting means includes means for selecting an effective server connection based on a connection history of said client and present connection conditions.

16. The system of Claim 15, wherein said selecting means includes means for accessing said connection history from a

table of server connection parameters, which are utilized to determine an effective route.

17. The system of Claim 16, wherein said dynamically connecting means includes means for evaluating said server connection parameters for each of a plurality of servers to determine said effective connection route relative to all other possible routes within said connection type.

18. The system of Claim 17, wherein said dynamically connecting means further includes means for encoding routing information about said effective connection route in a connection protocol of said client.

19. The system of Claim 18, wherein said encoding means includes means for including a call-back mechanism in said connection protocol, wherein relevant connection information, including one or more of said connection parameters, is returned to said client for updating said table.

20. The system of Claim 13, wherein said client is equipped with multiple connection media and said dynamically connecting means includes means for selecting one of said

multiple connection media to complete said connection request.

21. The system of Claim 20, wherein said selecting means includes means for selecting a connecting media, which provides the effective connection route.

22. A computer program product for providing a client with a connection to a network, said program product comprising:

a computer readable medium;

program instructions stored on said computer readble medium for:

selecting a connection type from a plurality of connection types; and

in response to a receipt of a connection request, dynamically connecting said client to a selected server of said network based on a determination of an effective route for completing said connection request, given said selected connection type.

23. The computer program product of Claim 22, wherein said program instructions for said selecting step includes program instructions for providing a graphical user interface with selectable options for each of said plurality of connection types, in response to a user request to configure said client with one of said plurality of connection types.

24. The computer program product of Claim 23, wherein said program instructions for said selecting step includes

program instructions for selecting an effective server connection based on a connection history of said client and present connection conditions.

25. The computer program product of Claim 24, wherein said program instructions for said selecting step includes program instructions for accessing said connection history from a table of server connection parameters, which are utilized to determine an effective connection route.

26. The computer program product of Claim 25, wherein said program instructions for said dynamically connecting step further includes program instructions for encoding routing information about said effective connection route in a connection protocol of said client.

27. The computer program product of Claim 26, wherein said encoding means includes means for including a call-back mechanism in said connection protocol, wherein relevant connection information, including one or more of said connection parameters, is returned to said client for updating said table.

28. The computer program product of Claim 27, wherein said client is equipped with multiple connection media and said

program instructions for said dynamically connecting step s includes program instructions for selecting one of said multiple connection media, which provides the effective connection route, to complete said connection request.

29. A graphical user interface of a browser running on a computer system, comprising:

a first set of user selectable buttons representing a plurality of connection media, said buttons having a first display characteristic indicating when a functionality associated with each of said first set of user selectable buttons is presently available, a second display characteristic indicating when said functionality is not available, and a third display characteristic indicating when one of said first set of user selectable buttons has been selected; and

a second set of user selectable buttons representing a user preference for server connections including a default server selection, an override default selection, and an automatic routing selection.

ABSTRACT OF THE DISCLOSURE

DYNAMICALLY AFFECTING BROWSER NETWORK COMMUNICATIONS PERFORMANCE

A method, system and program product for utilization within a client that provides an optimal connection between the client and a network. The method, system and program product provides a graphical user interface for receiving user selection and connection requests and a connection utility for connecting the client system with a selected server of the network. The selected server and connection route is dynamically selected from a plurality of servers in response to the receipt of a connection request, based on a determination of a best route for completing the connection request. The client is provided with multiple connection media and the determination of the best route includes a selection of the connection medium to utilize for the connection. Also, a table of connection information is stored within the client and utilized in evaluating which connection route is optimal.

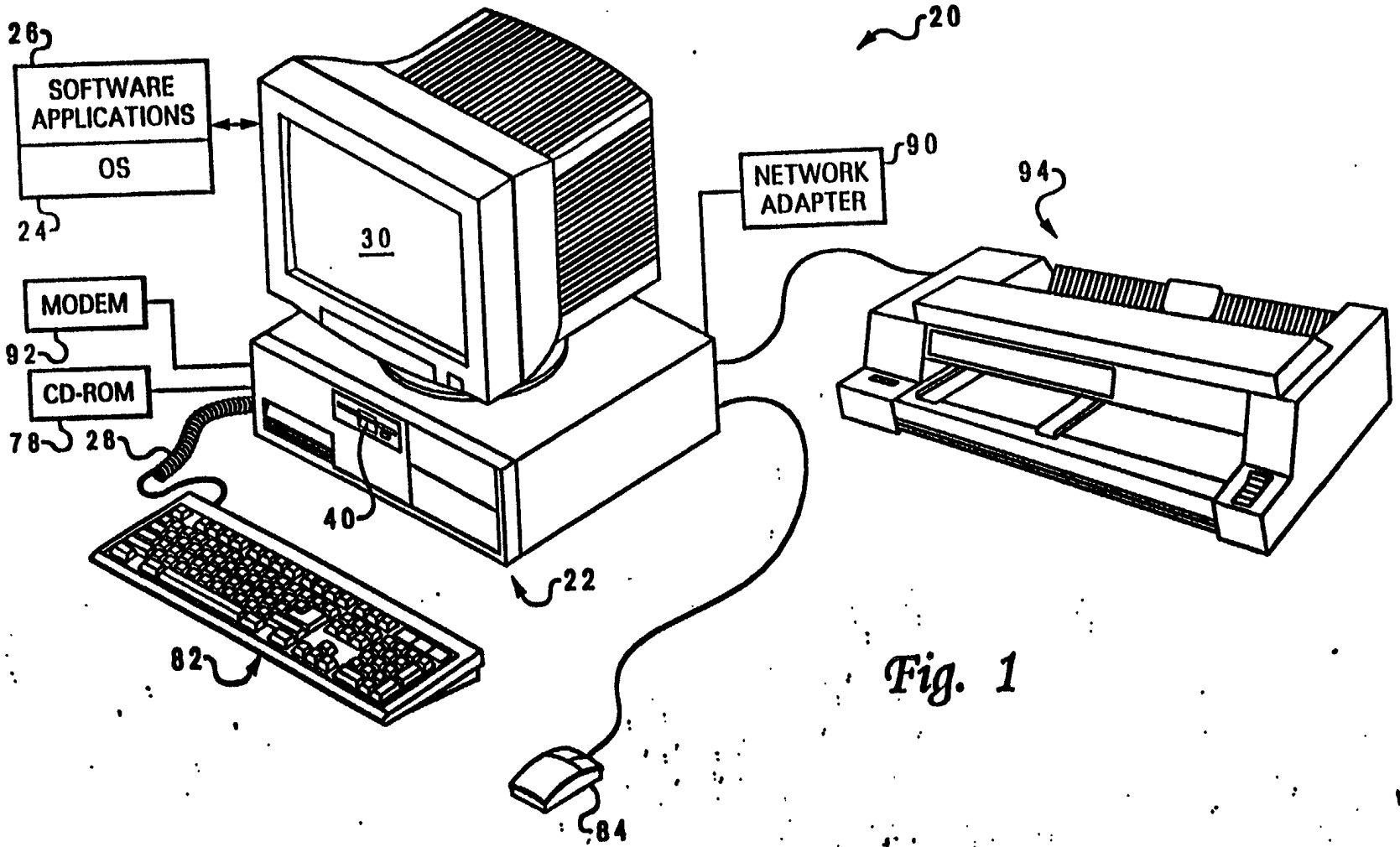
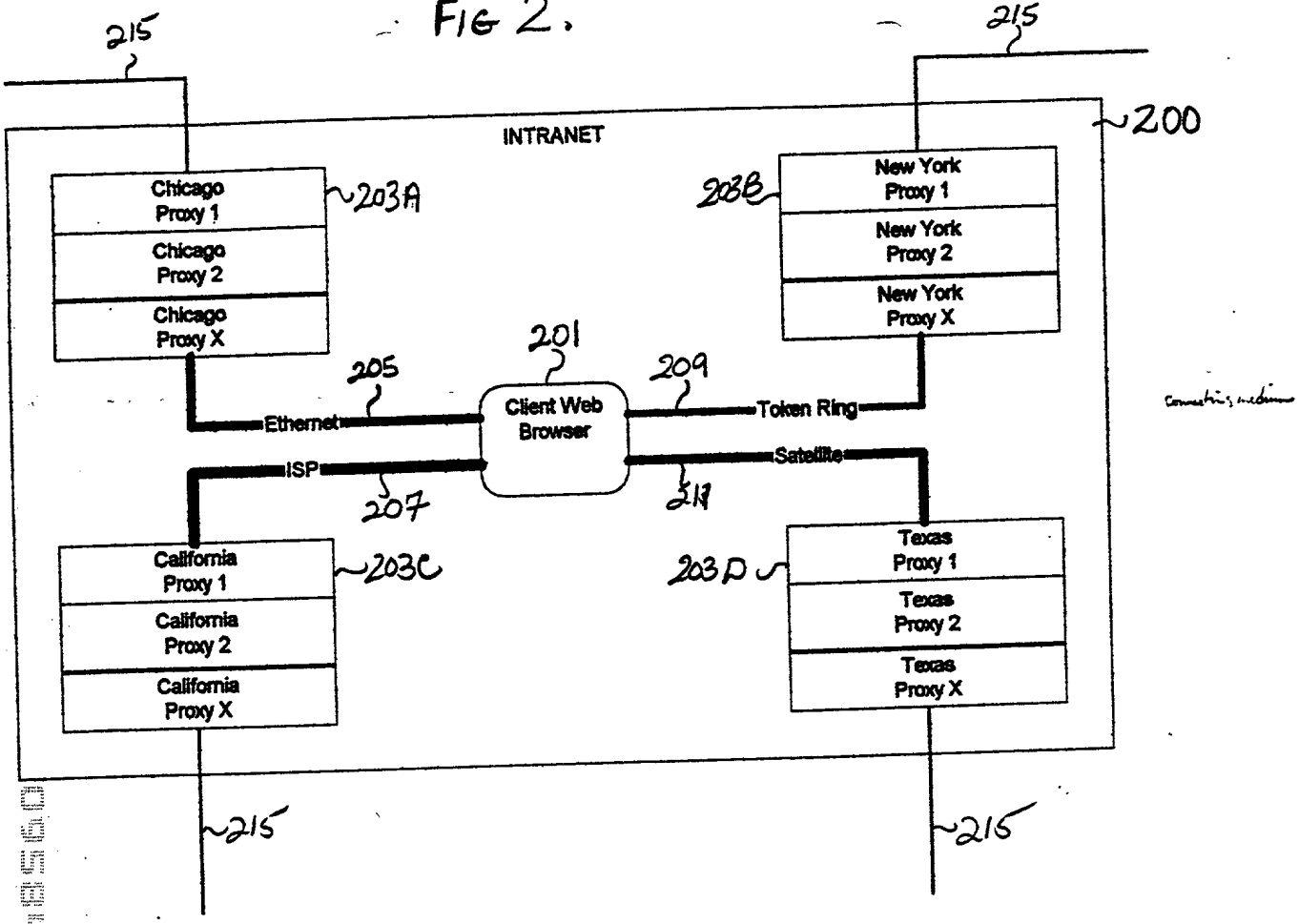

26
SOFTWARE APPLICATIONS
OS
24
30
20
NETWORK ADAPTER
90
94
MODEM
92
CD-ROM
78
28
40
22
82
84

Fig. 1

FIG 2.


215
215
INTRANET
200
Chicago Proxy 1
Chicago Proxy 2
Chicago Proxy X
203A
203B
New York Proxy 1
New York Proxy 2
New York Proxy X
201
205
209
Ethernet
Client Web Browser
Token Ring
ISP
207
Satellite
211
California Proxy 1
California Proxy 2
California Proxy X
203C
203D
Texas Proxy 1
Texas Proxy 2
Texas Proxy X
215
215

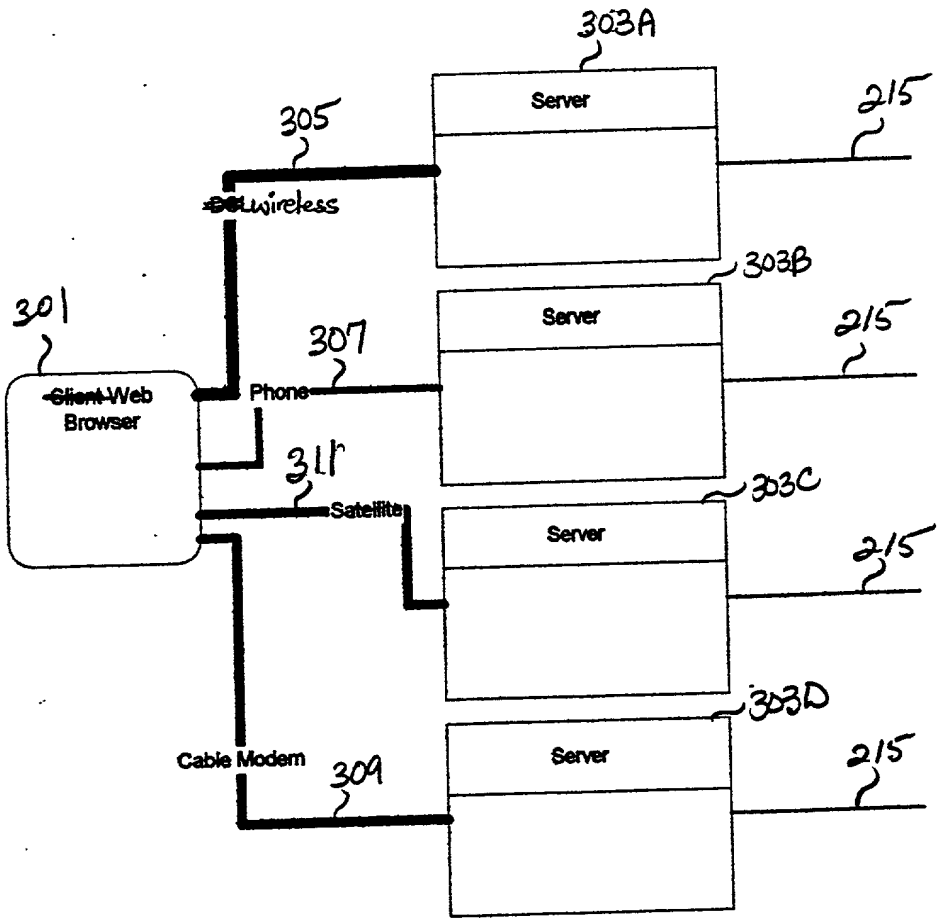

303A
Server
215
305
Wireless
303B
301
Server
215
307
Client Web Browser
Phone
311
303C
Satellite
Server
215
303D
Cable Modem
309
Server
215

FIG. 3

FIG. 4

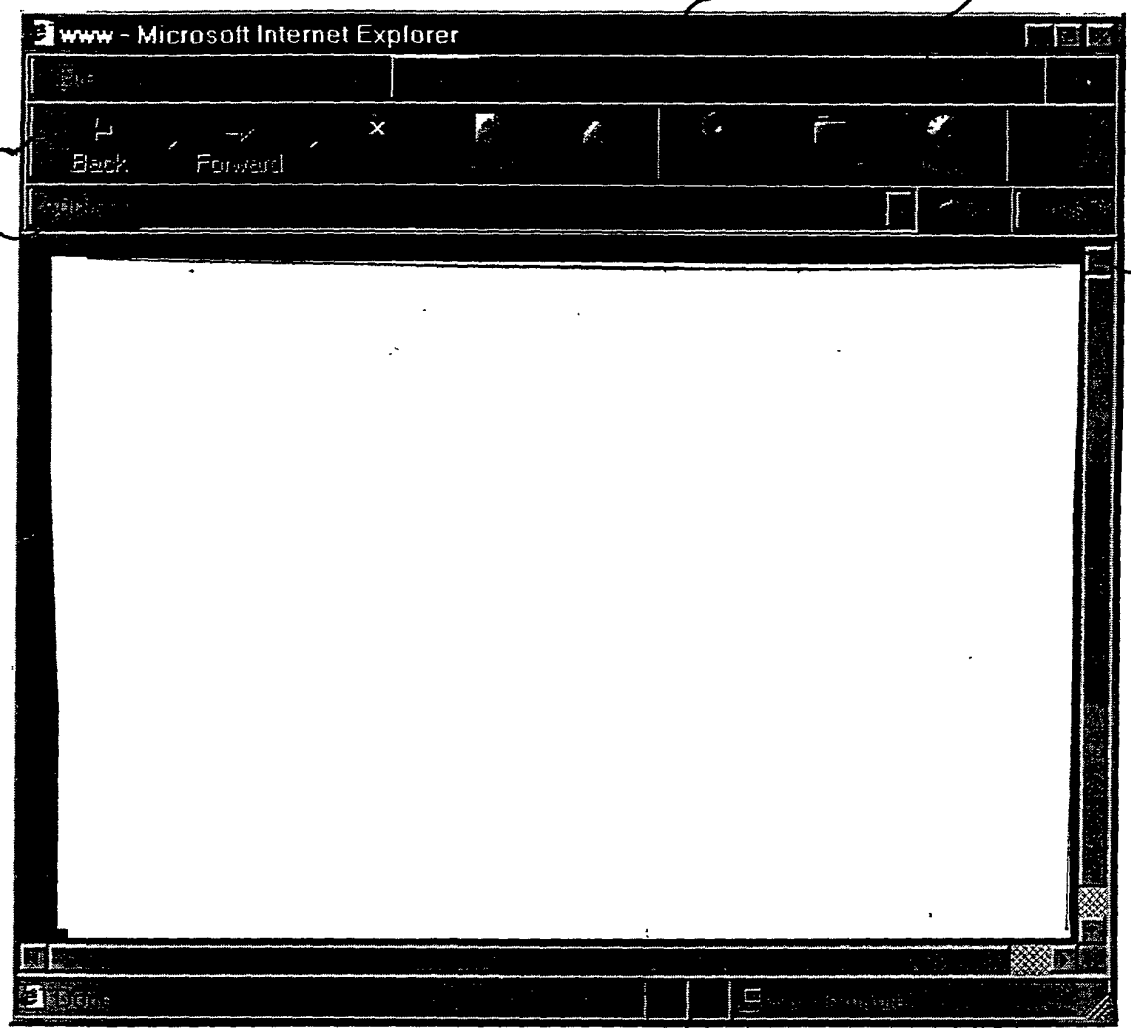

400
407
www - Microsoft Internet Explorer
401
Back
Forward
403
405

FIG. 5A

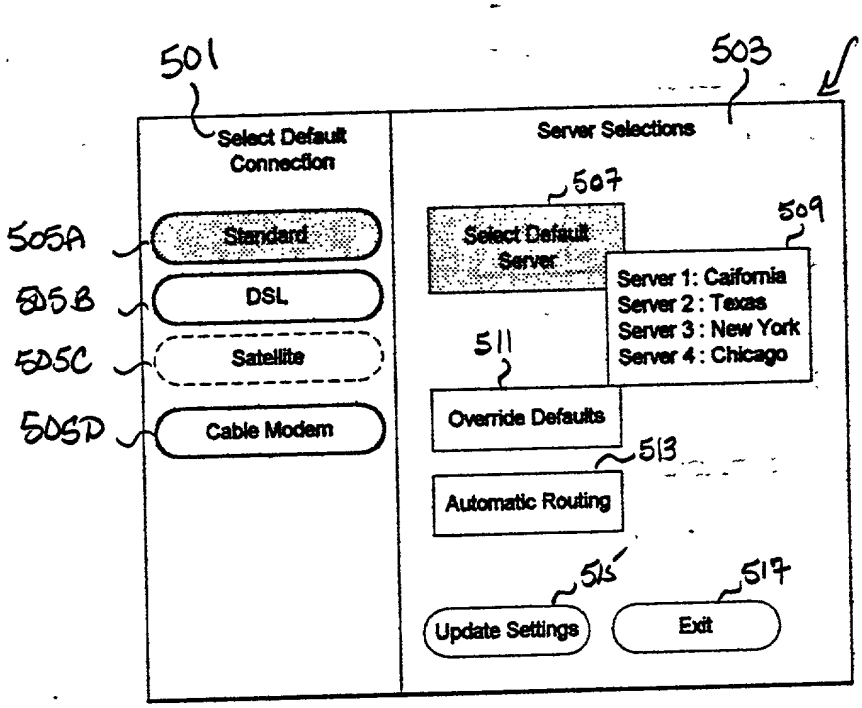

500
501
503
Select Default Connection
Server Selections
505A
Standard
505B
DSL
505C
Satellite
505D
Cable Modem
507
Select Default Server
509
Server 1: California
Server 2: Texas
Server 3: New York
Server 4: Chicago
511
Override Defaults
513
Automatic Routing
515
Update Settings
517
Exit

FIG 5B

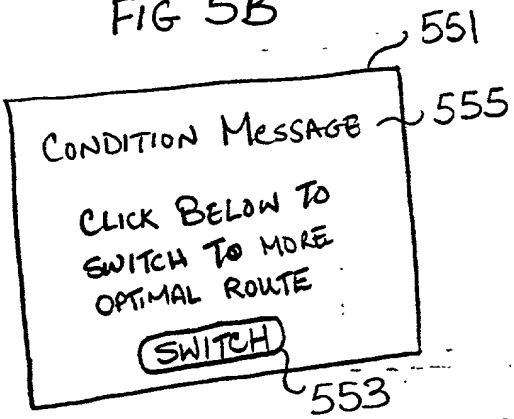

551
CONDITION MESSAGE
555
CLICK BELOW TO SWITCH TO MORE OPTIMAL ROUTE
SWITCH
553

Fig. 6

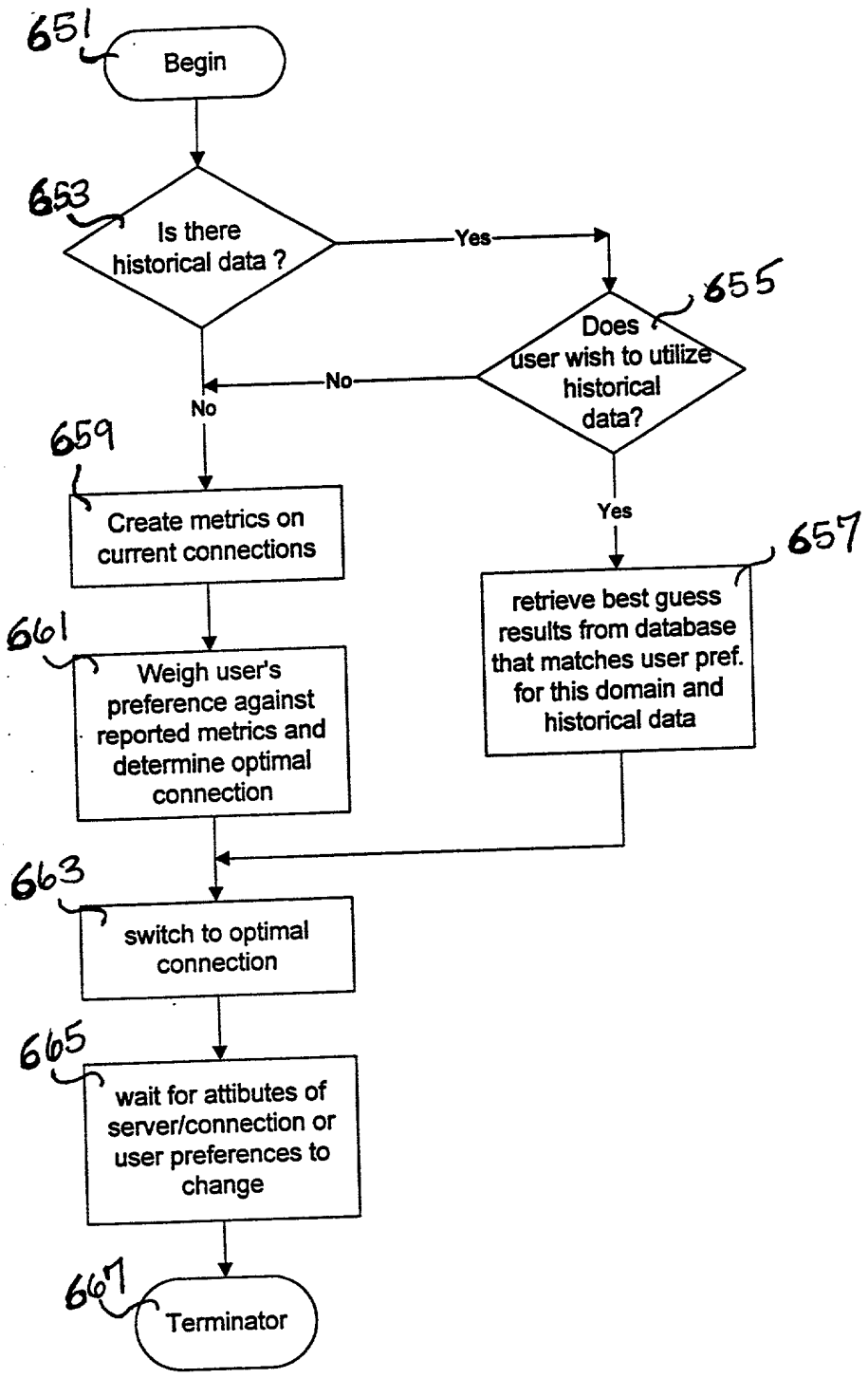

651
Begin
653
Is there historical data ?
Yes
No
655
Does user wish to utilize historical data?
No
Yes
659
Create metrics on current connections
657
retrieve best guess results from database that matches user pref. for this domain and historical data
661
Weigh user's preference against reported metrics and determine optimal connection
663
switch to optimal connection
665
wait for attibutes of server/connection or user preferences to change
667
Terminator

FIG 7

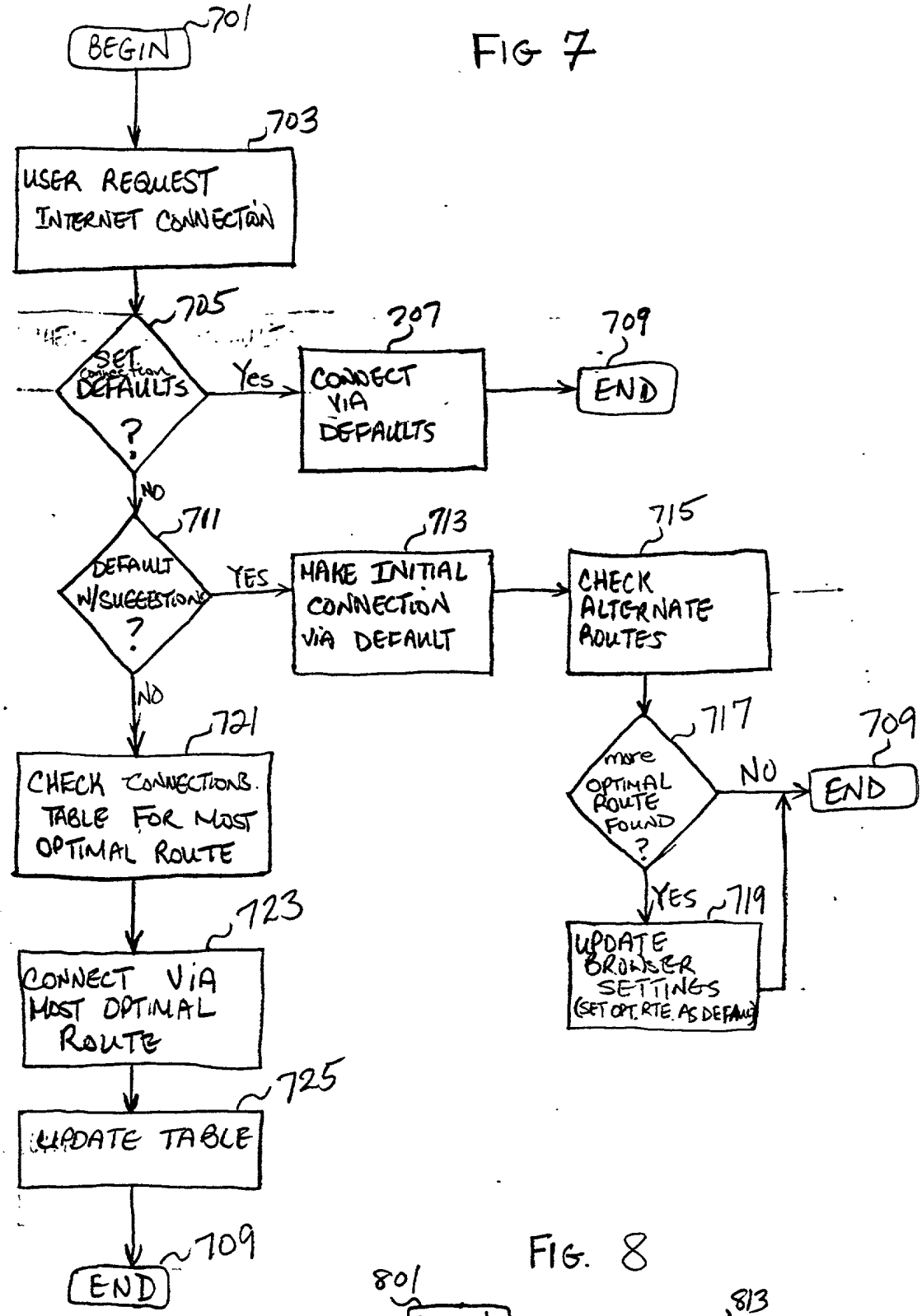

701
BEGIN
703
USER REQUEST INTERNET CONNECTION
705
SET CONNECTION DEFAULTS ?
Yes
707
CONNECT VIA DEFAULTS
709
END
NO
711
DEFAULT W/SUGGESTIONS ?
YES
713
MAKE INITIAL CONNECTION VIA DEFAULT
715
CHECK ALTERNATE ROUTES
NO
721
CHECK CONNECTIONS TABLE FOR MOST OPTIMAL ROUTE
717
MORE OPTIMAL ROUTE FOUND ?
NO
709
END
YES
719
UPDATE BROWSER SETTINGS (SET OPT. RTE. AS DEFAULT)
723
CONNECT VIA MOST OPTIMAL ROUTE
725
UPDATE TABLE
709
END

FIG. 8

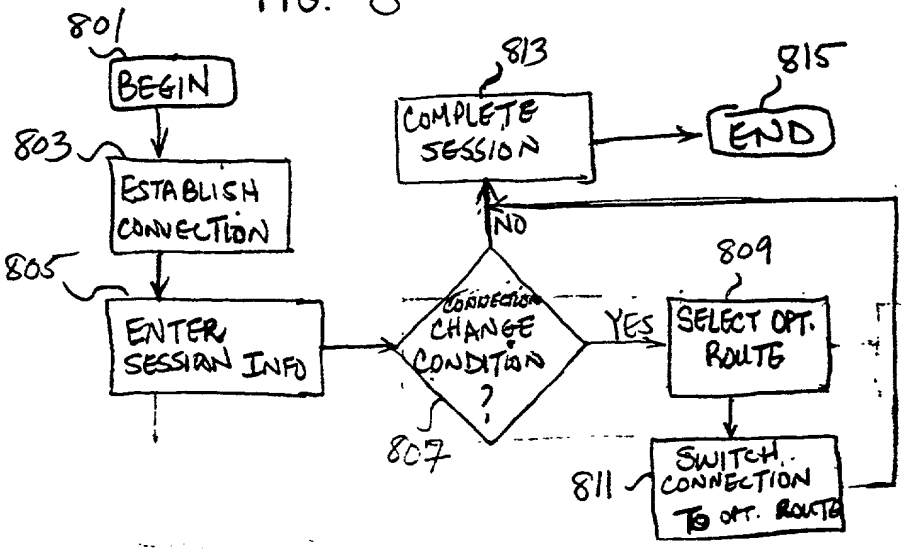

801
BEGIN
803
ESTABLISH CONNECTION
805
ENTER SESSION INFO
CONNECTION CHANGE CONDITION ?
807
YES
809
SELECT OPT. ROUTE
811
SWITCH CONNECTION TO OPT. ROUTE
NO
813
COMPLETE SESSION
815
END

FIG. 9

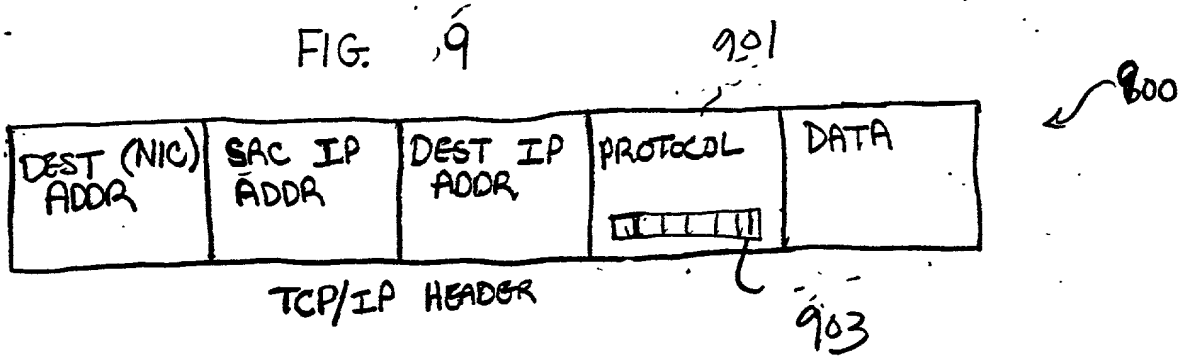

901
900
DEST (NIC) ADDR
SRC IP ADDR
DEST IP ADDR
PROTOCOL
DATA
TCP/IP HEADER
903

FIG. 11

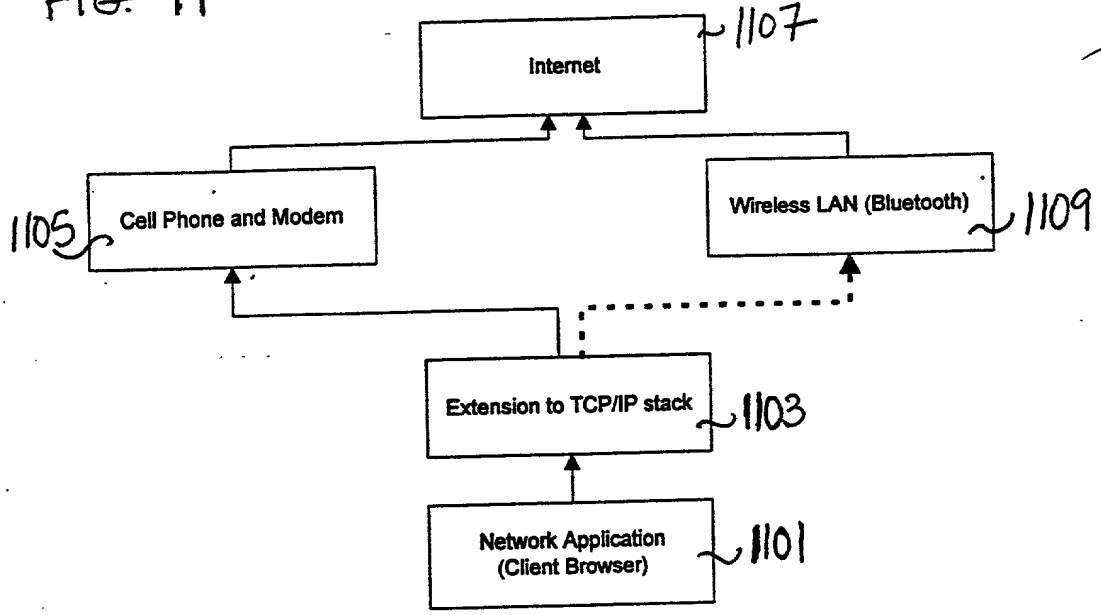
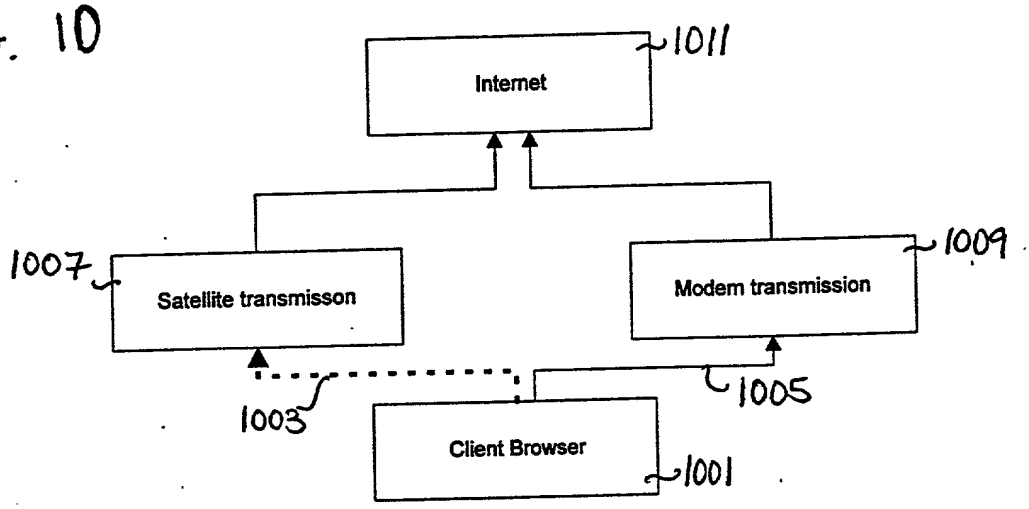

Internet
1107
Cell Phone and Modem
1105
Wireless LAN (Bluetooth)
1109
Extension to TCP/IP stack
1103
Network Application
(Client Browser)
1101

FIG. 10


Internet
1011
1007
Satellite transmisson
Modem transmission
1009
1003
1005
Client Browser
1001

DOCKET NUMBER: AUS000123US1

DECLARATION AND POWER OF ATTORNEY FOR

PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

DYNAMICALLY AFFECTING BROWSER NETWORK COMMUNICATIONS PERFORMANCE

the specification of which (check one)

X is attached hereto.
__ was filed on _____
as Application Serial No. _____
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):			Priority Claimed
_____	_____	_____	___ Yes___ No
(Number)	(Country)	(Day/Month/Year)	

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior

application and the national or PCT international filing date of this application:

(Application Serial #)	(Filing Date)	(Status)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John W. Henderson, Jr., Reg. No. 26,907; Thomas E. Tyson, Reg. No. 28,543; Robert M. Carwell, Reg. No. 28,499; Jeffrey S. LaBaw, Reg. No. 31,633; Douglas H. Lefeve, Reg. No. 26,193; Casimer K. Salys, Reg. No. 28,900; David A. Mims, Jr., Reg. No. 32,708; Volel Emile, Reg. No. 39,969; James H. Barksdale, Jr. Reg. No. 24,091; Anthony V. S. England, Reg. No. 35,129; Leslie A. Van Leeuwen, Reg. No. 42,196; Marilyn S. Dawkins, Reg. No. 31,140; Mark E. McBurney, Reg. No. 33,114; Christopher A. Hughes, Reg. No. 26,914; John E. Hoel, Reg. No. 26,279; Joseph C. Redmond, Jr., Reg. No. 18,753; Andrew J. Dillon, Reg. No. 29,634; Daniel E. Venglarik, Reg. No. 39,409; Jack V. Musgrove, Reg. No. 31,986; Brian F. Russell, Reg. No. 40,796; Steven Lin, Reg. No. 35,250; Matthew W. Baca, Reg. No. 42,277; Antony P. Ng, Reg. No. 43,427; John G. Graham, Reg. No. 19,563; Matthew S. Anderson, Reg. No. 39,093; Michael R. Barre, Reg. No. 44,023; Andrew Mitchell Harris, Reg. No. 42,638; Richard N. McCain, Reg. No. 43,785; Michael E. Noe, Jr., Reg. No. 44,975; and Sidney L. Weatherford, Reg. No. P-45,602.

Send correspondence to: Andrew J. Dillon, FELSMAN, BRADLEY, VADEN, GUNTER & DILLON, LLP, Suite 350 Lakewood on the Park, 7600B North Capital of Texas Highway, Austin, Texas 78731, and direct all telephone calls to Andrew J. Dillon, (512) 343-6116.

FULL NAME OF SOLE OR FIRST INVENTOR: BRUCE A. BEADLE

INVENTORS SIGNATURE: _____ DATE: 5-23-2000

RESIDENCE: 3648 FLORA VISTA LOOP
ROUND ROCK, TEXAS 78681

CITIZENSHIP: U.S.A.

POST OFFICE ADDRESS: 3648 FLORA VISTA LOOP
ROUND ROCK, TEXAS 78681

DOCKET NUMBER: AUS000123US1

FULL NAME OF SECOND INVENTOR: MICHAEL WAYNE BROWN

INVENTORS SIGNATURE: Michael Wayne Brown DATE: 5/23/2000

RESIDENCE: 529 RIVER DOWN ROAD
GEORGETOWN, TEXAS 78628

CITIZENSHIP: U.S.A.

POST OFFICE ADDRESS: 529 RIVER DOWN ROAD
GEORGETOWN, TEXAS 78628

FULL NAME OF THIRD INVENTOR: RANDOLPH MICHAEL FORLENZA

INVENTORS SIGNATURE: Randolph Michael Forlenza DATE: 5/25/00

RESIDENCE: 5807 STANDING ROCK DRIVE
AUSTIN, TEXAS 78730

CITIZENSHIP: U.S.A.

POST OFFICE ADDRESS: 5807 STANDING ROCK DRIVE
AUSTIN, TEXAS 78730

FULL NAME OF FOURTH INVENTOR: DOUGLAS SCOTT ROTHERT

INVENTORS SIGNATURE: Douglas Scott Rothert DATE: 5-23-2000

RESIDENCE: 11901 HOBBY HORSE CT., APT. 1815
AUSTIN, TEXAS 78758

CITIZENSHIP: U.S.A.

POST OFFICE ADDRESS: 11901 HOBBY HORSE CT., APT. 1815
AUSTIN, TEXAS 78758

FULL NAME OF FIFTH INVENTOR: MIGUEL SANG

INVENTORS SIGNATURE: Miguel Sang DATE: 5/25/2000

RESIDENCE: ~~3401 PARMER LANE, APT. 628~~ ~~AUSTIN, TEXAS 78727~~ 5817 MIRAMONTE DR AUSTIN, TX 78759 MS

CITIZENSHIP: ~~U.S.A.~~ DOMINICAN REPUBLIC MS

POST OFFICE ADDRESS: ~~3401 PARMER LANE, APT. 628~~ ~~AUSTIN, TEXAS 78727~~ MS